

# Network-based clustering with mixtures of $\ell_1$ -penalized Gaussian graphical models: an empirical investigation

Steven Hill\*

Centre for Complexity Science and  
Department of Statistics  
University of Warwick  
Coventry, UK  
s.hill@nki.nl

Sach Mukherjee

The Netherlands Cancer Institute  
Amsterdam, The Netherlands  
s.mukherjee@nki.nl

January 11, 2013

## Abstract

In many applications, multivariate samples may harbor previously unrecognized heterogeneity at the level of conditional independence or network structure. For example, in cancer biology, disease subtypes may differ with respect to subtype-specific interplay between molecular components. Then, both subtype discovery and estimation of subtype-specific networks present important and related challenges. To enable such analyses, we put forward a mixture model whose components are sparse Gaussian graphical models. This brings together model-based clustering and graphical modeling to permit simultaneous estimation of cluster assignments and cluster-specific networks. We carry out estimation within an  $\ell_1$ -penalized framework, and investigate several specific penalization regimes. We present empirical results on simulated data and provide general recommendations for the formulation and use of mixtures of  $\ell_1$ -penalized Gaussian graphical models.

**Keywords:** Model-based clustering; Network structure inference; Gaussian graphical models;  $\ell_1$ -regularization; Mixture models

---

\*Current Address: The Netherlands Cancer Institute, Amsterdam, The Netherlands

# 1 Introduction

Clustering of high-dimensional data has been the focus of much statistical research over the past decade. The increasing prevalence of high-throughput biological data has been an important motivation for such efforts. In molecular biology applications, clustering can be used to either group variables together (e.g. to find sets of co-regulated genes; Eisen *et al.*, 1998; Toh and Horimoto, 2002), group samples together (e.g. to discover disease subtypes characterized by similar gene expression profiles; Golub *et al.*, 1999; Alizadeh *et al.*, 2000), or to simultaneously group both variables together and samples together ('bi-clustering' methods; Alon *et al.*, 1999; Madeira and Oliveira, 2004). In this work we focus on the second of these approaches; that is, to cluster a small-to-moderate number of high-dimensional samples. Numerous clustering algorithms have been used in biological applications, notably for gene expression data; see Datta and Datta (2003); Thalamuthu *et al.* (2006); Kerr *et al.* (2008) and de Souto *et al.* (2008) for reviews and comparisons of various methods, including K-means, hierarchical clustering and model-based clustering. Model-based clustering (Fraley and Raftery, 1998; McLachlan *et al.*, 2002) with Gaussian mixture models is a popular approach to clustering that is rooted in an explicit statistical model.

Another area that has received much attention in recent years is structural inference for graphical models. In a graphical model, a graph, comprising vertices and linking edges, is used to describe probabilistic relationships between variables; structural inference refers to estimation of the graph edge structure. In bioinformatics applications, structural inference is important for the elucidation of molecular networks, such as gene regulatory or protein signaling networks, from biochemical data. Many methods for structural inference have been proposed in the literature, including those based on Bayesian networks (Friedman *et al.*, 2000; Husmeier, 2003; Segal *et al.*, 2003; Needham *et al.*, 2007; Mukherjee and Speed, 2008; Hill *et al.*, 2012) and Gaussian graphical models (Schäfer and Strimmer, 2005; Toh and Horimoto, 2002; Friedman *et al.*, 2008). They are reviewed, along with other approaches, in Lee and Tzou (2009); Heckler *et al.* (2009) and Markowitz and Spang (2007).

In this paper, we develop a model-based clustering approach with components defined by graphical models. This allows simultaneous recovery of cluster assignments and estimation of cluster-specific graphical model structure. Our work is of particular relevance to questions concerning undiscovered heterogeneity at the level of network structure. Such questions arise in diverse molecular biology applications. The edge structure of biological networks can differ depending on context, e.g. disease state or other subtype, in ways that may have implications for targeted and personalized therapies (Pe'er and Hachohen, 2011). When such heterogeneity is well-understood, samples can be partitioned into suitable subsets prior to network inference (Altay *et al.*, 2011) (or other supervised network-based approaches (Chuang *et al.*, 2007)). However, in practice, molecular classifications that underpin such stratifications may be uncertain and moreover hitherto unknown subtypes may be present. In the latter case, subtype identification is itself of independent interest, as in the context of many diseases, including cancer. A crucial observation is that if subtypes differ with respect to underlying network structure, clustering and network inference become coupled tasks. Clustering methods that do not model cluster-specific network structure (including K-means, hierarchical clustering or model-based clustering with diagonal covariance matrices (de Souto *et al.*, 2008)), may lead to cluster assignments that do not reflect the underlying biology and that may also compromise the ability to elucidate network structure. Equally, structural inference based on the full, unclustered data can be severely confounded by the data heterogeneity.

As cluster-specific network models, we use sparse Gaussian graphical models. These are multivariate Gaussian models in which an undirected graph is used to represent conditional independence relationships between variables. Inferring the edge set of a Gaussian graphical model is equivalent to identifying the location of non-zero entries in the precision matrix (see Section 2.1 below for details). There is a rich literature on precision matrix estimation in the context of sparse Gaussian graphical models, with the seminal paper by Dempster (1972) proposing sparse estimation by setting entries in the precision matrix to zero. Edwards (2000) provides a review of standard approaches, such as greedy stepwise backward selection, for identifying zero entries in the precision matrix. More recent approaches have focused on using regularization, and  $\ell_1$  penalization in particular, to achieve sparsity. Meinshausen and Bühlmann (2006) use  $\ell_1$ -penalized regression (lasso; Tibshirani, 1996) to perform neighborhood selection for each node in the graph. A sparse precision matrix can subsequently be obtained via constrained maximum likelihood estimation using the inferred sparse graph structure. Maximum penalized likelihood estimators with an  $\ell_1$  penalty applied to the precision matrix have been proposed by Yuan and Lin (2007); Friedman *et al.* (2008); Rothman *et al.* (2008) and D’Aspremont *et al.* (2008). Analogous to the lasso, where sparse models are encouraged by shrinking some regression coefficients to be exactly zero, the  $\ell_1$  penalty on the precision matrix encourages sparsity by estimating some matrix entries as exactly zero. Since a sparse precision matrix corresponds to a sparse Gaussian graphical model structure,  $\ell_1$  penalized estimation is well-suited for inference of molecular networks, where sparsity is often a valid assumption. Moreover, regularization enables estimation in the challenging ‘large  $p$ , small  $n$ ’ regime that is ubiquitous in these settings, but renders standard covariance estimators inapplicable or ill-behaved.

Our work adds to the literature in two main ways. First, the penalized mixture-model formulation we propose extends previous work. Mukherjee and Hill (2011) put forward a related ‘network clustering’ approach, but this is not rooted in a formal statistical model and estimation is carried out using a heuristic, K-means-like algorithm with ‘hard’ cluster assignments. We show empirically that likelihood-based inference via an EM formulation confers benefits over this approach. EM algorithms for penalized likelihoods have previously been proposed for finite mixture of regression models (Khalili and Chen, 2007; Städler *et al.*, 2010) and for penalized model-based clustering (Pan and Shen, 2007; Zhou *et al.*, 2009). The approach in Zhou *et al.* (2009) is similar to the one here. However, our  $\ell_1$  penalty takes a more general form, allowing also for dependence on mixing proportions at the level of the full likelihood. We show that at smaller sample sizes in particular, the  $\ell_1$  penalty we propose offers substantial gains. Furthermore, while we are interested in both clustering and cluster-specific network estimation, Zhou *et al.* (2009) focus on the use of variable selection to improve clustering accuracy.

Second, we present empirical results investigating the performance of penalisation regimes. A penalty parameter controls the extent to which sparsity is encouraged in the precision matrix and corresponding graphical model. The choice of method for setting the penalty parameter together with the different forms of the  $\ell_1$  penalty itself result in several possible regimes that can be difficult to choose between *a priori*. Our results show that the choice of regime can be influential and suggest general recommendations.

The remainder of this paper is organized as follows. In the next Section we introduce  $\ell_1$ -penalized estimation for Gaussian graphical models and model-based clustering, and then go on to describe the proposed mixture model. In Section 3 we present an empirical comparison, on synthetic data, of several regimes for the  $\ell_1$  penalty term and tuning parameter selection. In Section 4 we close with a discussion of our findings and suggest areas for future work.

## 2 Methods

### 2.1 Penalized estimation of Gaussian graphical model structure

Let  $\mathbf{X} = (X_1, \dots, X_p)^\top$  denote a random vector having  $p$ -dimensional Gaussian density  $f(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  with mean  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$ . A Gaussian graphical model uses an undirected graph  $G = (V, E)$  to describe conditional independence relationships between the random variables  $X_1, \dots, X_p$ . The  $p$  vertices  $V$  of the graph are identified with  $X_1, \dots, X_p$  with edge  $(i, j) \notin E$  if and only if  $X_i$  is conditionally independent of  $X_j$  given all other variables, or equivalently, if and only if there is zero partial correlation between  $X_i$  and  $X_j$  given all other variables ( $\rho_{ij} = 0$ ). Let  $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$  denote the inverse covariance or precision matrix, and let  $\omega_{ij}$  be entry  $(i, j)$  of  $\boldsymbol{\Omega}$ . Then, the relationship between  $\boldsymbol{\Omega}$  and partial correlations is given by  $\rho_{ij} = -\frac{\omega_{ij}}{\sqrt{\omega_{ii}\omega_{jj}}}$ . Therefore, non-zero entries in  $\boldsymbol{\Omega}$  correspond to edges in the GGM, that is  $\omega_{ij} \neq 0 \iff (i, j) \in E$ . Thus, inferring the edge set of a GGM is equivalent to identifying the location of non-zero entries in the precision matrix.

Suppose  $\mathbf{x}_1, \dots, \mathbf{x}_n$ , with  $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^\top$  is a random sample from  $f(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ . Let  $\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$  denote sample mean and  $\hat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$  sample covariance. The precision matrix  $\boldsymbol{\Omega}$  may be estimated by maximum likelihood. The log-likelihood function is given, up to a constant, by

$$l(\boldsymbol{\Omega}) = \log |\boldsymbol{\Omega}| - \text{tr}(\boldsymbol{\Omega} \hat{\boldsymbol{\Sigma}}) \quad (1)$$

where  $|\cdot|$  and  $\text{tr}(\cdot)$  denote matrix determinant and trace respectively. The maximum likelihood estimate is given by inverting the sample covariance matrix,  $\hat{\boldsymbol{\Omega}} = \hat{\boldsymbol{\Sigma}}^{-1}$ . However for  $n < p$ ,  $\hat{\boldsymbol{\Sigma}}$  is singular and so cannot be used to estimate  $\boldsymbol{\Omega}$ . Even when  $n \geq p$ ,  $\hat{\boldsymbol{\Omega}}$  can be a poor estimator for large  $p$  and does not in general yield sparse precision matrices.

Sparse estimates can be encouraged by placing an  $\ell_1$  penalty on the entries of the precision matrix  $\boldsymbol{\Omega}$ . This results in the following penalized log-likelihood:

$$l_p(\boldsymbol{\Omega}) = \log |\boldsymbol{\Omega}| - \text{tr}(\boldsymbol{\Omega} \hat{\boldsymbol{\Sigma}}) - \lambda \|\boldsymbol{\Omega}\|_1 \quad (2)$$

where  $\|\boldsymbol{\Omega}\|_1 = \sum_{i,j} |\omega_{i,j}|$  is the elementwise  $\ell_1$  matrix norm and  $\lambda$  is a non-negative tuning parameter controlling sparsity of the estimate. The maximum penalized likelihood estimate is obtained by maximizing (2) over symmetric, positive-definite matrices. This is a convex optimization problem and several procedures have been proposed to obtain solutions. Yuan and Lin (2007) used the maxdet algorithm, while D'Aspremont *et al.* (2008) proposed a more efficient semi-definite programming algorithm using interior point optimization. Rothman *et al.* (2008) offered a fast approach employing Cholesky decomposition and the local quadratic approximation, and Friedman *et al.* (2008) proposed the even faster graphical lasso algorithm, based on the coordinate descent algorithm for the lasso. We use the graphical lasso algorithm in our investigations and refer the interested reader to the references for full details.

### 2.2 Gaussian mixture models

We now suppose  $\mathbf{x}_1, \dots, \mathbf{x}_n$  is a random sample from a finite Gaussian mixture distribution,

$$f(\mathbf{x}_i; \boldsymbol{\Theta}) = \sum_{k=1}^K \pi_k f_k(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (3)$$

where the mixing proportions  $\pi_k$  satisfy  $0 \leq \pi_k \leq 1$  and  $\sum_{k=1}^K \pi_k = 1$ ,  $f_k$  is the  $p$ -dimensional multivariate Gaussian density with component-specific mean  $\boldsymbol{\mu}_k$  and covariance  $\boldsymbol{\Sigma}_k$ , and  $\boldsymbol{\Theta} = \{(\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) : k = 1, \dots, K\}$  is the set of all unknown parameters. The log-likelihood for the sample is given by

$$l(\boldsymbol{\Theta}) = \sum_{i=1}^n \log \left( \sum_{k=1}^K \pi_k f_k(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right). \quad (4)$$

Maximizing this log-likelihood is difficult due to its non-convexity. The Expectation-Maximization (EM) algorithm (Dempster *et al.*, 1977) can be used to obtain maximum likelihood estimates.

In model-based clustering (Fraley and Raftery, 1998; McLachlan *et al.*, 2002), each mixture component corresponds to a cluster. In the present setting, since each cluster (or component) is Gaussian distributed with a cluster-specific (unconstrained) covariance matrix, each cluster represents a distinct Gaussian graphical model.

### 2.3 Mixture of penalized Gaussian graphical models

In the Gaussian mixture model with cluster-specific covariance matrices, the number of parameters is of order  $Kp^2$ . Estimation is more challenging than for a single precision matrix (or Gaussian graphical model) and so, as described above, in settings where number of variables  $p$  is moderate-to-large in relation to sample size  $n$ , overfitting and invalid covariance estimates are a concern. We employ an  $\ell_1$  penalty on each of the  $K$  precision matrices to promote sparsity and ameliorate these issues. Such  $\ell_1$  penalties have previously been proposed for clustering with Gaussian graphical models (Zhou *et al.*, 2009; Mukherjee and Hill, 2011).

We propose the following penalized log-likelihood,

$$l_p(\boldsymbol{\Theta}) = \sum_{i=1}^n \log \left( \sum_{k=1}^K \pi_k f_k(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right) - \frac{n}{2} p_{\lambda, \gamma}(\boldsymbol{\Theta}) \quad (5)$$

where the penalty term is given by

$$p_{\lambda, \gamma}(\boldsymbol{\Theta}) = \lambda \sum_{k=1}^K \pi_k^\gamma \|\boldsymbol{\Omega}_k\|_1 \quad (6)$$

and  $\gamma$  is a binary parameter controlling the form of the penalty term. Setting  $\gamma = 0$  results in the conventional penalty term, as used in Zhou *et al.* (2009), with no dependence on the mixing proportions  $\pi_k$ . Setting  $\gamma = 1$  weights the penalty from each cluster by its corresponding mixing proportion. While this form of penalty is novel in this setting, an analogous penalty has been proposed by Khalili and Chen (2007) and Städler *et al.* (2010) for  $\ell_1$ -penalized finite mixture of regression models. In this work, we empirically compare these two forms of penalty term for clustering with, and estimation of, Gaussian graphical models.

### 2.4 Maximum penalized likelihood

As with the unpenalized log-likelihood (4), the penalized likelihood (5) can be maximized using an EM algorithm, which we now describe. Our algorithm is similar to that of Zhou *et al.* (2009), but they consider only the  $\gamma = 0$  regime and also penalize the mean vectors to perform variable selection.

Let  $z_i$  be a latent variable satisfying  $z_i = k$  if observation  $\mathbf{x}_i$  belongs to cluster  $k$ . Then we have  $P(z_i = k) = \pi_k$  and  $p(\mathbf{x}_i | z_i = k) = f_k(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ . The penalized log-likelihood for the complete data  $\{\mathbf{x}_i, z_i\}_{i=1}^n$  is

$$l_{p,c}(\boldsymbol{\Theta}) = \sum_{i=1}^n \log(\pi_{z_i}) + \log(f_{z_i}(\mathbf{x}_i | \boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i})) - \frac{n}{2} p_{\lambda, \gamma}(\boldsymbol{\Theta}). \quad (7)$$

In the E-step of the EM, given current estimates of the parameters  $\boldsymbol{\Theta}^{(t)}$ , we compute

$$\begin{aligned} Q(\boldsymbol{\Theta} | \boldsymbol{\Theta}^{(t)}) &= \mathbb{E} \left[ l_{p,c}(\boldsymbol{\Theta}) | \{\mathbf{x}_i\}_{i=1}^n, \boldsymbol{\Theta}^{(t)} \right] \\ &= \sum_{i=1}^n \sum_{k=1}^K \tau_{ik}^{(t)} [\log(\pi_k) + \log(f_k(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k))] - \frac{n}{2} p_{\lambda, \gamma}(\boldsymbol{\Theta}) \end{aligned} \quad (8)$$

where  $\tau_{ik}^{(t)}$  is the posterior probability of observation  $\mathbf{x}_i$  belonging to cluster  $k$ ,

$$\tau_{ik}^{(t)} = \frac{\pi_k^{(t)} f_k(\mathbf{x}_i | \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)})}{\sum_{j=1}^K \pi_j^{(t)} f_j(\mathbf{x}_i | \boldsymbol{\mu}_j^{(t)}, \boldsymbol{\Sigma}_j^{(t)})} \quad (9)$$

and can be thought of as a ‘soft’ cluster assignment.

In the M-step we seek to maximize  $Q(\boldsymbol{\Theta} | \boldsymbol{\Theta}^{(t)})$  with respect to  $\boldsymbol{\Theta}$  to give new estimates for the parameters  $\boldsymbol{\Theta}^{(t+1)}$ . When  $\gamma = 0$  the mixture proportions  $\pi_k$  do not appear in the penalty term  $p_{\lambda, \gamma}(\boldsymbol{\Theta})$  and so we use the following standard EM update for unpenalized Gaussian mixture models:

$$\pi_k^{(t+1)} = \frac{\sum_{i=1}^n \tau_{ik}^{(t)}}{n}. \quad (10)$$

For  $\gamma = 1$ , since  $\pi_k$  appears in the penalty term, maximization of  $Q(\boldsymbol{\Theta} | \boldsymbol{\Theta}^{(t)})$  with respect to  $\pi_k$  is non-trivial. We follow Khalili and Chen (2007) and use the standard update (10). If the standard update improves  $Q(\boldsymbol{\Theta} | \boldsymbol{\Theta}^{(t)})$  then this is sufficient to obtain (local) maxima of (5). An improvement is not guaranteed here, but as found in Khalili and Chen (2007), the method works well in practice.

Since the penalty term is independent of  $\boldsymbol{\mu}_k$ , we again use the standard update,

$$\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_{i=1}^n \tau_{ik}^{(t)} \mathbf{x}_i}{\sum_{i=1}^n \tau_{ik}^{(t)}}. \quad (11)$$

The update for  $\boldsymbol{\Sigma}_k$ , or equivalently  $\boldsymbol{\Omega}_k$ , is given by

$$\begin{aligned} \boldsymbol{\Omega}_k^{(t+1)} &= \arg \max_{\boldsymbol{\Omega}_k} \left[ \sum_{i=1}^n \tau_{ik}^{(t)} \left( \log |\boldsymbol{\Omega}_k| - \text{tr}(\boldsymbol{\Omega}_k \mathbf{S}_k^{(t)}) \right) - n\lambda \left( \pi_k^{(t+1)} \right)^\gamma \|\boldsymbol{\Omega}_k\|_1 \right] \\ &= \arg \max_{\boldsymbol{\Omega}_k} \left[ \log |\boldsymbol{\Omega}_k| - \text{tr}(\boldsymbol{\Omega}_k \mathbf{S}_k^{(t)}) - \tilde{\lambda}_k^{(t)} \|\boldsymbol{\Omega}_k\|_1 \right] \end{aligned} \quad (12)$$

where

$$\mathbf{S}_k^{(t)} = \frac{\sum_{i=1}^n \tau_{ik}^{(t)} (\mathbf{x}_i - \boldsymbol{\mu}_k^{(t+1)}) (\mathbf{x}_i - \boldsymbol{\mu}_k^{(t+1)})^\top}{\sum_{i=1}^n \tau_{ik}^{(t)}} \quad (13)$$

is the standard EM update for  $\Sigma$  and

$$\tilde{\lambda}_k^{(t)} = n\lambda \frac{\left(\pi_k^{(t+1)}\right)^\gamma}{\sum_{i=1}^n \tau_{ik}^{(t)}}. \quad (14)$$

The optimization problem in (12) is of the form of that in (2) with  $\hat{\Sigma}$  replaced by  $\mathbf{S}_k^{(t)}$  and a scaled tuning parameter  $\tilde{\lambda}_k^{(t)}$ . Hence we can use the efficient graphical lasso algorithm (Friedman *et al.*, 2008) to perform the optimization.

From (10) we have

$$\tilde{\lambda}_k^{(t)} = \begin{cases} \frac{\lambda}{\pi_k^{(t+1)}} & \text{if } \gamma = 0 \\ \lambda & \text{if } \gamma = 1 \end{cases} \quad (15)$$

Hence, when  $\gamma = 0$ ,  $\tilde{\lambda}_k^{(t)}$  is a cluster-specific parameter inversely proportional to effective cluster sample size, whereas  $\gamma = 1$  simply yields  $\lambda$ . We note that, even though  $\gamma = 1$  gives a cluster-specific tuning parameter in the penalized log-likelihood (5) while  $\gamma = 0$  does not, the converse is actually true for the EM updates (12).

Our overall algorithm is as follows:

1. Initialize  $\Theta^{(0)}$ : Randomly assign each observation  $\mathbf{x}_i$  into one of  $K$  clusters, subject to a minimum cluster size  $n_{\min}$ . Set  $\pi_k^{(0)} = n_k/n$  where  $n_k$  is the number of observations assigned to cluster  $k$ , set  $\mu_k^{(0)}$  to sample mean of cluster  $k$ , and set  $\Omega_k^{(0)}$  to the maximum penalized likelihood estimate for the cluster  $k$  precision matrix (using (2)).
2. E-step: Calculate posterior probabilities ('soft' assignments)  $\tau_{ik}^{(t)}$  using (9).
3. M-step: Calculate updated parameter estimates  $\Theta^{(t+1)}$  using (10)-(15).
4. Iterate or terminate: Increment  $t$ . Repeat steps 2 and 3, or stop if one of the following criteria is satisfied:
  - A maximum number of iterations  $T$  is reached;  $t > T$ .
  - A minimum cluster size  $n_{\min}$  is reached;  $\sum_{i=1}^n \tau_{ik}^{(t)} < n_{\min}$  for some  $k$ .
  - Relative change in penalized log-likelihood is below a threshold  $\epsilon$ ;  
 $|l_p(\Theta)^{(t)} / l_p(\Theta)^{(t-1)} - 1| \leq \epsilon$ .

In all experiments below we set  $T = 100$ ,  $n_{\min} = 4$  and  $\epsilon = 10^{-4}$ . Since the EM algorithm may only find local maxima, we perform 25 random restarts and select the one giving the highest penalized log-likelihood. 'Hard' cluster assignments are obtained by assigning observations to the cluster  $k$  with largest probability  $\tau_{ik}$ .

## 2.5 Tuning parameter selection

Two approaches are commonly used to set the tuning parameter: cross-validation (CV) and criteria such as BIC. In multifold CV, the data samples are partitioned into  $M$  data subsets, denoted by  $\mathbf{X}^{(m)}$  for  $m = 1, \dots, M$ . Let  $\hat{\Theta}_\lambda^{(-m)} = \{(\pi_{k\lambda}, \mu_{k\lambda}, \Sigma_{k\lambda}) : k = 1, \dots, K\}$  denote the penalized likelihood estimate, obtained using tuning parameter  $\lambda$  and by application of the EM algorithm

described above to all data save that in subset  $\mathbf{X}^{(m)}$  (training data). Performance of this estimate is assessed using the predictive log-likelihood; that is, Equation (4) applied to subset  $\mathbf{X}^{(m)}$  (test data). This is repeated  $M$  times, allowing each subset to play the role of test data. The CV score is

$$\text{CV}(\lambda) = \sum_{m=1}^M \sum_{i: \mathbf{x}_i \in \mathbf{X}^{(m)}} \log \left( \sum_{k=1}^K \hat{\pi}_{k\lambda}^{(-m)} f_k \left( \mathbf{x}_i \mid \hat{\boldsymbol{\mu}}_{k\lambda}^{(-m)}, \hat{\boldsymbol{\Sigma}}_{k\lambda}^{(-m)} \right) \right). \quad (16)$$

Then we choose  $\lambda$  that maximizes  $\text{CV}(\lambda)$ , where the maximization is performed via a grid search. Finally, the selected value is used to learn penalized likelihood estimates from all data.

In the larger sample case, an alternative to multifold CV is to partition the data into two and perform a single train/test iteration, selecting  $\lambda$  that maximizes the predictive log-likelihood on the test data with penalized parameter estimates from the training data.

We define the following BIC score for our penalized mixture model:

$$\text{BIC}(\lambda) = -2l(\hat{\boldsymbol{\Theta}}_\lambda) + \text{df}_\lambda \log(n) \quad (17)$$

where  $l(\cdot)$  is the unpenalized log-likelihood (4),  $\hat{\boldsymbol{\Theta}}_\lambda$  is the penalized likelihood estimate obtained with tuning parameter  $\lambda$  and  $\text{df}_\lambda$  is degrees of freedom. Yuan and Lin (2007) proposed an estimate of the degrees of freedom for  $\ell_1$ -penalized precision matrix estimation, which generalizes to our penalized Gaussian mixture model setting to give

$$\text{df}_\lambda = K(p+1) - 1 + \sum_{k=1}^K \# \left\{ (j, j') : j \leq j', (\hat{\omega}_{k\lambda})_{jj'} \neq 0 \right\}. \quad (18)$$

where  $(\hat{\omega}_{k\lambda})_{jj'}$  is element  $(j, j')$  in  $\hat{\boldsymbol{\Omega}}_{k\lambda}$ , the penalized likelihood estimate for the cluster  $k$  precision matrix, using tuning parameter  $\lambda$ . Using a grid search, we choose  $\lambda$  that minimizes  $\text{BIC}(\lambda)$ .

BIC is often preferred over CV as it is less computationally intensive. However, we note that, even BIC can be computationally expensive when used within clustering since each  $\lambda$  value in the grid search requires a full application of EM-based clustering. Hence, to reduce computation time, we also consider a heuristic, approximate version of these approaches. The heuristic we propose relies on the notion that the optimal tuning parameter value does not depend strongly on cluster assignments but rather largely on general properties of the data (such as  $p$  and  $n$ ). The approach proceeds as follows. First, observations are randomly assigned to clusters, producing  $K$  pseudo-clusters each with mean size  $n/K$ . Second, parameter estimates are obtained for the pseudo-clusters.  $\hat{\pi}_k$  is taken to be the proportion of samples in pseudo-cluster  $k$  and  $\hat{\boldsymbol{\mu}}_k$  is the sample mean of pseudo-cluster  $k$ . Then, for varying  $\lambda$ , we obtain penalized estimates  $\hat{\boldsymbol{\Omega}}_{k\lambda}$  by optimizing (2) for each pseudo-cluster with the graphical lasso. This can be done efficiently using the `glassopath` algorithm in R (Friedman *et al.*, 2008) which obtains penalized estimates for all considered values of  $\lambda$  simultaneously. Third, using these estimates, CV (BIC) scores are calculated and maximized (minimized) to select  $\lambda$ . These three steps are repeated multiple times and  $\lambda$  values obtained are averaged to produce a final value.

### 3 Simulated data

In this section we apply the  $\ell_1$ -penalized Gaussian graphical model clustering approach to simulated data. We consider a number of combinations of  $\ell_1$  penalty term and tuning parameter scheme



(as described in Methods above) and assess their performance in carrying out three related tasks. First, recovery of correct cluster assignments. Second, estimation of cluster-specific graphical model structure (i.e. location of non-zero entries in cluster-specific precision matrices). Third, estimation of cluster-specific precision matrices (i.e. estimation of matrix elements, not just locations of non-zero entries). We note that this latter task is of less interest here since we are mainly concerned with clustering and inference of cluster-specific network structure.

### 3.1 Data generation

In our simulation we considered  $p$ -dimensional data consisting of  $K = 2$  clusters, each with a known and distinct Gaussian graphical model structure (i.e. sparse precision matrix). Sparse precision matrices were created using an approach based on that used by Rothman *et al.* (2008) and Cai *et al.* (2011). In particular, we created a symmetric  $p \times p$  matrix  $B_1$  with zeros everywhere except for  $p$  randomly chosen pairs of symmetric, off-diagonal entries, which took value 0.5. A second matrix  $B_2$  was created from  $B_1$  by selecting half of the  $p$  non-zero symmetric pairs at random and relocating them to new randomly chosen symmetric positions. We then set  $\Omega_k = B_k + \delta_k I$ , where  $\delta_k$  is the minimal value such that  $\Omega_k$  is positive-definite with condition number less than  $p$ . Finally, the precision matrices  $\Omega_k$  were standardized to have unit diagonals. This resulted in cluster-specific Gaussian graphical models each with  $p$  edges, half of which were shared by both network structures. Data were generated from  $\mathcal{N}(\mathbf{0}, \Omega_1^{-1})$  and  $\mathcal{N}(\frac{\alpha}{\sqrt{p}}\mathbf{1}, \Omega_2^{-1})$  for clusters 1 and 2 respectively, where  $\mathbf{1}$  is the vector of ones. The mean of cluster two is defined such that the parameter  $\alpha$  sets the Euclidean distance between the cluster means. In the experiments below we consider  $p = 25, 50, 100$  and cluster sample sizes of  $n_k = 15, 25, 50, 100, 200$ . We set  $\alpha = 3.5$ , resulting in individual component-wise means for cluster two of 0.70, 0.50 and 0.35 for  $p = 25, 50$  and 100 respectively. This reflects the challenging scenario where clusters do not have substantial differences in mean values, but display heterogeneity in network structure while also sharing some network structure across clusters.

### 3.2 Methods and regimes

We assessed ability to recover correct cluster assignments from 50 simulated datasets, under the following four regimes for the penalty term  $p_{\lambda, \gamma}(\Theta)$  in (6):  $\gamma = 0$  or 1 and  $\lambda$  set by BIC or a train/test scheme, maximizing the predictive log-likelihood on an independent test dataset with cluster sample sizes matching the training dataset. These regimes are described fully above and summarized in Table 1. We also compared with (i) K-means; (ii) standard non-penalized full-covariance Gaussian mixture models estimated using EM; and (iii) ‘network clustering’, an  $\ell_1$ -penalized Gaussian graphical model clustering approach proposed by Mukherjee and Hill (2011). This is similar to the approach employed here but uses a heuristic, K-means-like algorithm with ‘hard’ cluster assignments rather than a mixture-model formulation with EM. For (i) we used the `kmeans` function in the MATLAB statistics toolbox with  $K=2$  and 1000 random initializations and for (iii) we used MATLAB function `network_clustering` (Mukherjee and Hill, 2011). For (ii) and (iii) we used the same stopping criteria as described in Methods above (namely,  $T = 100$ ,  $n_{\min} = 4$  and  $\epsilon = 10^{-4}$ ) and again carried out 25 random restarts. Method (iii) requires maximization of  $K$  penalized log-likelihoods of form (2) above (one for each cluster). For setting penalty parameters for this method, we considered either a single tuning parameter  $\lambda$  shared across both clusters and set by BIC or train/test, or cluster-specific tuning parameters  $\lambda_k$ , set analytically

**Table 1:** Clustering methods and regimes investigated, with corresponding abbreviations.

Method	Penalty term	Tuning parameter selection	Abbrev.
mixture of $\ell_1$ -penalized Gaussian graphical models with EM ('soft' assignments)	$p_{\lambda,\gamma}(\Theta)$ with $\gamma = 0$ : $\lambda \sum_{k=1}^K \ \Omega_k\ _1$	Train/test	T0
		BIC	B0
	$p_{\lambda,\gamma}(\Theta)$ with $\gamma = 1$ : $\lambda \sum_{k=1}^K \pi_k \ \Omega_k\ _1$	Train/test	T1
		BIC	B1
$\ell_1$ -penalized Gaussian graphical models ('hard' assignments) Mukherjee and Hill (2011)	$\lambda \ \Omega_k\ _1, k = 1, \dots, K$	Train/test	Th
		BIC	Bh
	$\lambda_k \ \Omega_k\ _1, k = 1, \dots, K$	Analytic <sup>1</sup>	Ah
K-means	n/a	n/a	KM
non-penalized Gaussian mixture model with EM	n/a	n/a	NP

<sup>1</sup> following Banerjee *et al.* (2008) (see text for details)

before each call to the penalized estimator using the equation proposed by Banerjee *et al.* (2008, Equation 3). All computations were carried out in MATLAB R2010a, making an external call to the R package `glasso` (Friedman *et al.*, 2008). Table 1 gives abbreviations for all methods and regimes investigated, which are used below and in figures.

### 3.3 Tuning parameter selection

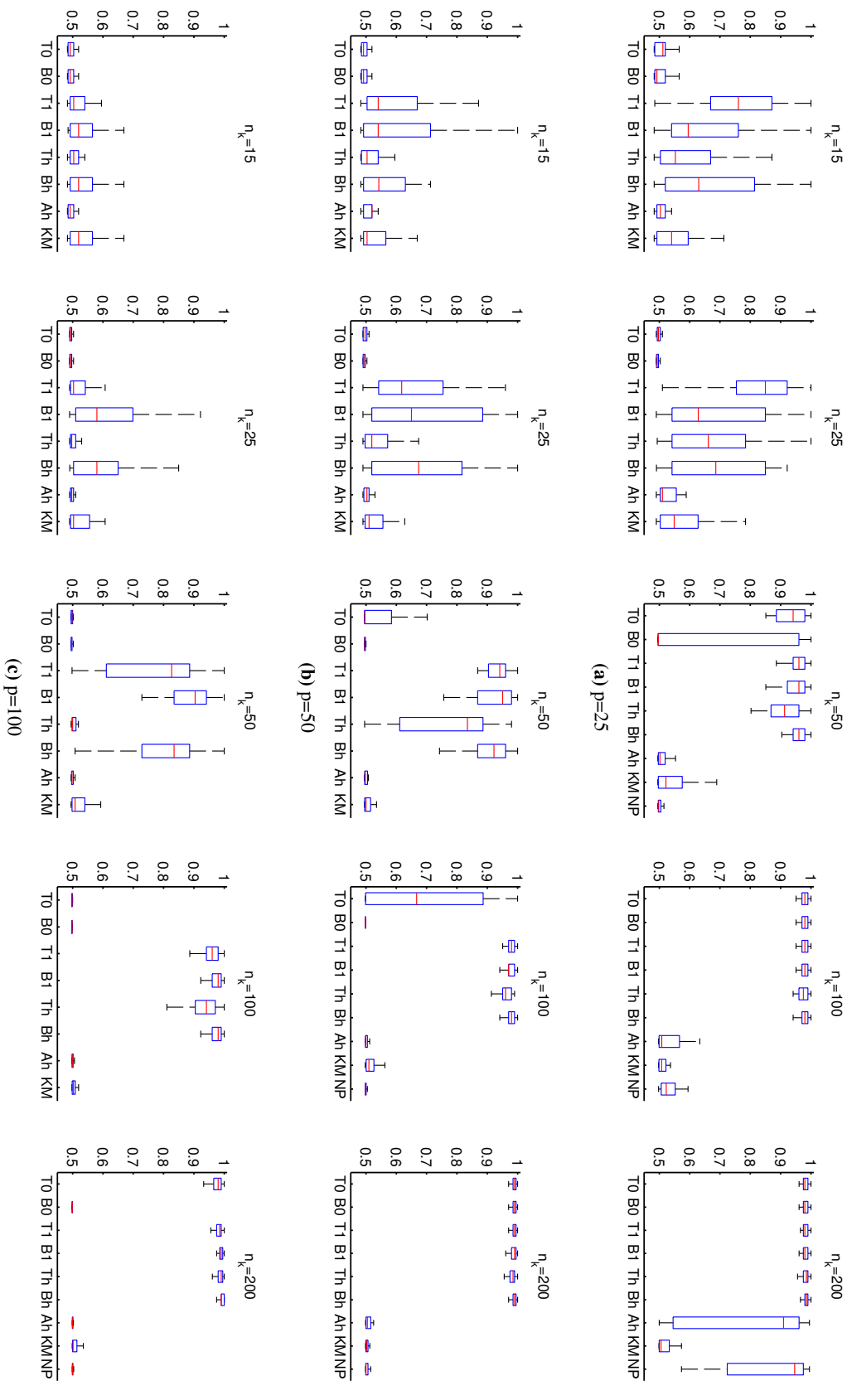
Table 2 shows average tuning parameter values selected by each regime. A grid search was used over values between 0.05 and 1.5, with increments of 0.05. Since, for  $\gamma = 0$ , the EM update tuning parameters  $\tilde{\lambda}_k$  in (12) differ from  $\lambda$ , we also show  $\tilde{\lambda}_k$  for these regimes. Using BIC to set the tuning parameter results in higher values than with train/test and, as expected,  $\lambda$  values increase with  $p$  and decrease with  $n_k$ .

### 3.4 Cluster assignment

Figure 1 shows Rand indices (with respect to the true cluster assignments) obtained from clustering the simulated data. The Rand index is a measure of similarity between cluster assignments, taking values between 0 and 1 (0 indicates complete disagreement and 1 complete agreement). Box plots are shown over 50 simulated datasets for each  $(p, n_k)$  regime. The  $\ell_1$ -penalized mixture model regimes with  $\gamma = 1$  in the penalty term (T1/B1) consistently provide the best clustering results. At the largest sample sizes both train/test (T1) and BIC (B1) offer good clustering performance, with high Rand indices reported. However, for smaller sample sizes, train/test outperforms BIC at the lowest data dimensionality ( $p = 25$ ), while the converse is true at higher dimensions ( $p = 50, 100$ ). The non-mixture  $\ell_1$ -penalized method (Th/Bh) also performs well, but the corresponding mixture model approaches with  $\gamma = 1$  (T1/B1) are, for the most part, more effective at smaller sample sizes (see e.g.  $n_k = 50, p = 50, 100$ ). This difference in performance is likely due to a combination of

**Table 2:** Simulated data: tuning parameter values selected for the methods and regimes in Table 1. Three data dimensions ( $p = 25, 50, 100$ ) and five per-cluster sample sizes ( $n_k = 15, 25, 50, 100, 200$ ) were considered. For the  $\gamma = 0$  regimes (T0/B0) EM update tuning parameters  $\hat{\lambda}_k$ , given in (15), are also shown. (Results shown are mean values over 50 simulated datasets for each  $(p, n_k)$  regime; standard deviations given in parentheses; cluster-specific tuning parameter  $\lambda_1$  and EM update parameter  $\tilde{\lambda}_1$  correspond to the largest cluster.)

p	n <sub>k</sub>	λ	T <sup>0</sup>		λ	B <sup>0</sup>		λ(=λ̃ <sub>k</sub> )	T <sup>1</sup>		λ(=λ̃ <sub>k</sub> )	B <sup>1</sup>		λ	Th		λ	Bh		λ <sub>1</sub>	Ah	
			λ̃ <sub>1</sub>	λ̃ <sub>2</sub>		λ̃ <sub>1</sub>	λ̃ <sub>2</sub>		λ(=λ̃ <sub>k</sub> )	λ(=λ̃ <sub>k</sub> )		λ	λ		λ	λ						
25	15	0.25 (0.07)	0.31 (0.07)	1.61 (0.73)	0.57 (0.11)	0.67 (0.12)	4.05 (0.88)	0.43 (0.08)	0.91 (0.15)	0.44 (0.11)	0.98 (0.16)	1.82 (0.43)	3.80 (1.54)									
	25	0.17 (0.04)	0.20 (0.04)	1.84 (0.82)	0.48 (0.07)	0.53 (0.08)	5.53 (0.86)	0.33 (0.05)	0.70 (0.11)	0.35 (0.10)	0.76 (0.14)	1.39 (0.23)	3.12 (1.42)									
	50	0.10 (0.01)	0.18 (0.04)	0.62 (0.94)	0.26 (0.13)	0.31 (0.10)	5.29 (3.99)	0.19 (0.03)	0.39 (0.06)	0.25 (0.03)	0.40 (0.07)	1.14 (0.43)	2.01 (1.15)									
	100	0.05 (0.00)	0.10 (0.00)	0.10 (0.00)	0.08 (0.04)	0.14 (0.05)	0.72 (2.40)	0.11 (0.02)	0.23 (0.03)	0.13 (0.03)	0.23 (0.03)	0.81 (0.13)	1.14 (0.58)									
50	200	0.05 (0.00)	0.10 (0.00)	0.10 (0.00)	0.06 (0.02)	0.12 (0.04)	0.12 (0.04)	0.08 (0.02)	0.15 (0.01)	0.08 (0.02)	0.15 (0.01)	0.61 (0.08)	0.70 (0.12)									
	15	0.32 (0.06)	0.38 (0.06)	2.24 (0.61)	0.66 (0.09)	0.77 (0.10)	4.65 (0.72)	0.51 (0.08)	1.03 (0.12)	0.48 (0.09)	1.12 (0.13)	2.06 (0.38)	4.81 (2.16)									
	25	0.23 (0.04)	0.26 (0.04)	2.56 (0.74)	0.53 (0.06)	0.58 (0.06)	6.06 (0.76)	0.38 (0.08)	0.81 (0.10)	0.37 (0.09)	0.84 (0.11)	1.63 (0.31)	4.39 (1.42)									
	50	0.14 (0.03)	0.16 (0.02)	2.60 (1.53)	0.36 (0.03)	0.38 (0.03)	8.21 (1.16)	0.29 (0.02)	0.53 (0.08)	0.34 (0.06)	0.53 (0.06)	1.18 (0.18)	4.32 (1.78)									
100	100	0.07 (0.02)	0.10 (0.02)	1.32 (1.89)	0.25 (0.03)	0.25 (0.02)	10.50 (2.26)	0.15 (0.00)	0.30 (0.02)	0.19 (0.02)	0.30 (0.02)	0.90 (0.35)	3.06 (1.70)									
	200	0.05 (0.00)	0.10 (0.00)	0.10 (0.00)	0.05 (0.01)	0.10 (0.01)	0.10 (0.01)	0.10 (0.00)	0.20 (0.00)	0.10 (0.00)	0.20 (0.00)	0.67 (0.36)	1.84 (1.38)									
	15	0.39 (0.04)	0.45 (0.05)	2.69 (0.38)	0.72 (0.06)	0.84 (0.07)	5.04 (0.58)	0.55 (0.06)	1.20 (0.12)	0.55 (0.06)	1.23 (0.11)	2.70 (1.04)	5.30 (1.91)									
	25	0.28 (0.03)	0.31 (0.03)	3.07 (0.51)	0.57 (0.05)	0.63 (0.06)	6.54 (0.83)	0.42 (0.06)	0.96 (0.12)	0.41 (0.04)	0.99 (0.09)	2.22 (0.68)	6.50 (3.07)									
200	50	0.19 (0.02)	0.20 (0.02)	4.28 (0.74)	0.40 (0.04)	0.42 (0.04)	8.46 (1.56)	0.37 (0.05)	0.59 (0.04)	0.28 (0.04)	0.61 (0.05)	1.47 (0.35)	5.11 (1.60)									
	100	0.10 (0.01)	0.11 (0.01)	4.04 (1.07)	0.27 (0.02)	0.27 (0.02)	11.30 (1.86)	0.21 (0.02)	0.36 (0.02)	0.27 (0.02)	0.37 (0.03)	1.04 (0.24)	5.75 (2.75)									
	200	0.05 (0.00)	0.10 (0.00)	0.10 (0.00)	0.20 (0.01)	0.20 (0.01)	15.48 (2.60)	0.11 (0.02)	0.25 (0.01)	0.15 (0.00)	0.25 (0.01)	0.78 (0.20)	4.58 (2.40)									



**Figure 1:** Simulated data; cluster assignment results. Boxplots over the Rand index, a measure of similarity between inferred and true cluster labels (higher scores indicate better agreement, with a score of unity indicating perfect agreement), are shown for the methods and regimes in Table 1 at varying data dimensions  $p$  and per-cluster sample sizes  $n_k$ . (Results shown are over 50 simulated datasets for each  $(p, n_k)$  regime (see text for details); abbreviations for methods are summarized in Table 1; the non-penalized approach (NP) could not be used for  $n_k \leq p$  due to small sample sizes resulting in invalid covariance estimates.)

differences in tuning parameter (Supplementary Table S1) and less accurate parameter estimation for the non-mixture approaches because they do not take uncertainty of assignment into account. Interestingly, the mixture model with conventional penalty term ( $\gamma = 0$ ; T0/B0) shows poor performance relative to  $\gamma = 1$  except at larger sample sizes, with consistently poor clustering accuracy for  $n_k \leq p$ . Similar performance is observed for the non-mixture method with analytic tuning parameter selection (Ah). The poor performance of these three regimes appears to be related to the fact that they all use cluster-specific tuning parameters ( $\lambda_k$  for Ah and  $\hat{\lambda}_k$  within EM for T0/B0), resulting in considerable differences in cluster-level penalties (see Supplementary Table S1). We comment further on this finding in Discussion below. Due to its inability to capture the cluster-specific covariance (network) structure, K-means does not perform well, even at the largest sample size. Conventional non-penalized mixture models did not yield valid covariance estimates for sample sizes  $n_k \leq p$ , and for  $n_k > p$  we only observe gains relative to K-means in the large sample  $p = 25, n_k = 200$  case.

### 3.5 Estimation of graphical model structure

Figure 2 shows results for estimation of cluster-specific network structures for the methods and regimes in Table 1. For K-means, clustering is followed by an application, to each inferred cluster, of  $\ell_1$ -penalized precision matrix estimation (see (2)) with tuning parameter set by either BIC or train/test.

Ability to reconstruct cluster-specific networks is assessed by calculating the true positive rate (TPR), false positive rate (FPR) and Matthews Correlation Coefficient (MCC),

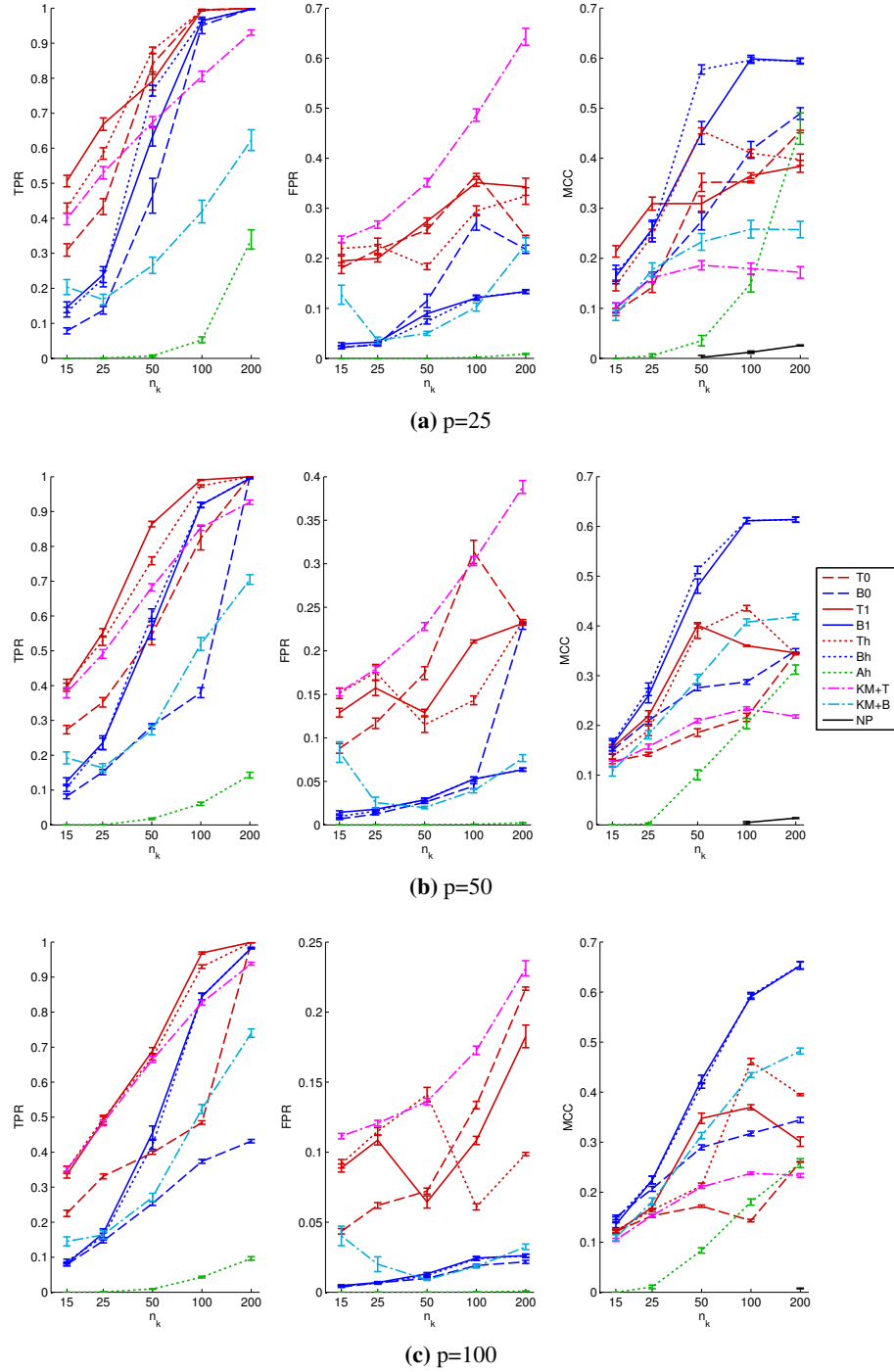
$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN} \quad (19)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (20)$$

where  $TP$ ,  $TN$ ,  $FP$  and  $FN$  denote the number of true positives, true negatives, false positives and false negatives (with respect to edges) respectively. MCC summarizes these four quantities into one score and is regarded as a balanced measure; it takes values between -1 and 1, with higher values indicating better performance (see e.g. Baldi *et al.* (2000) for further details). Since the convergence threshold in the `glasso` algorithm is  $10^{-4}$ , we take entries  $\hat{\omega}_{ij}$  in estimated precision matrices to be non-zero if  $|\hat{\omega}_{ij}| > 10^{-3}$ . Since cluster assignments can only be identified up to permutation, in all cases labels were permuted to maximize agreement with true cluster assignments before calculating these quantities.

Figure 2 shows MCC plotted against per-cluster sample size  $n_k$  and Supplementary Figure S1 shows corresponding plots for TPR and FPR. Due to selection of smaller tuning parameter values, BIC discovers fewer non-zeroes in the precision matrices than train/test, resulting in both fewer true positives and false positives. Under MCC, BIC, with either the  $\gamma = 1$  mixture model (B1) or the non-mixture approach (Bh), leads to the best network reconstruction (except at small sample sizes with  $p = 25$ ) and outperforms all other regimes at larger sample sizes.

In general, train/test is not competitive relative to BIC; at larger sample sizes the best train/test regimes (T1/Th) are only comparable with the worst performing BIC regimes (B0/KM+B). We note that the non-penalized mixture approach (NP), with sample size sufficiently large to provide valid covariance estimates, does not yield sparse precision matrices (MCC scores are approximately zero).



**Figure 2:** Simulated data; estimation of graphical model structure. True Positive Rate (TPR), False Positive Rate (FPR) and Matthews Correlation Coefficient (MCC) are shown as a function of per-cluster sample size  $n_k$  for the methods and regimes in Table 1 and at data dimensions  $p = 25, 50, 100$ . MCC is a balanced measure for classification performance, taking values between -1 and 1 with higher values indicating better agreement between true and inferred networks (see text for details). K-means clustering was followed by  $\ell_1$ -penalized estimation of Gaussian graphical model structure with penalty parameter set by train/test ('KM+T') or BIC ('KM+B'). (Mean values shown over 50 simulated datasets for each  $(p, n_k)$  regime, error bars show standard errors; non-penalized approach (NP) is only shown for MCC and could not be used for  $n_k \leq p$  due to small sample sizes resulting in invalid covariance estimates.)

### 3.6 Precision matrix estimation

We also assessed ability to accurately estimate underlying cluster-specific precision matrices (i.e. values of the matrix elements rather than only locations of non-zeros). Accuracy is assessed using the elementwise  $\ell_1$  norm,  $\sum_{k=1}^K \left\| \hat{\Omega}_k - \Omega_k \right\|_1$ , with inferred clusters matched to true clusters as described above. Results are shown in Table 3. In contrast to clustering and Gaussian graphical model estimation, where BIC regimes B1/Bh mainly provide the best performance, the train/test methods T1/Th are mostly similar or better than B1/Bh for precision matrix estimation (the exception being small  $n_k$ , higher  $p$  settings). Due to poor clustering performance, the mixture model approach with  $\gamma = 0$  does not perform well unless  $n_k$  is sufficiently large. Neither K-means clustering (followed by  $\ell_1$ -penalized precision matrix estimation), the penalized non-mixture approach with analytic tuning parameter selection (Ah), nor the non-penalized approach (NP) perform well, even at the largest sample size.

### 3.7 Approximate tuning parameter selection

We applied the heuristic method for setting the tuning parameter, described in Methods above, to the overall best-performing mixture model approach (regime B1;  $\gamma = 1$ , BIC). Figure 3 compares average  $\lambda$  values obtained using the heuristic method with those resulting from the full approach; we also show average Rand indices and computational timings. The  $\lambda$  values obtained via the heuristic scheme are well-behaved in the sense that they increase with  $p$  and decrease for larger  $n_k$ . We observe some bias relative to the full approach as the values obtained from the heuristic method are consistently higher. However, Rand indices remain in reasonable agreement and the heuristic offers some substantial computational gains; e.g. for  $p = 25$  we see reduction of about 90% in computation time. This suggests that the heuristic approach could be useful for fast, exploratory analyses.

## 4 Discussion

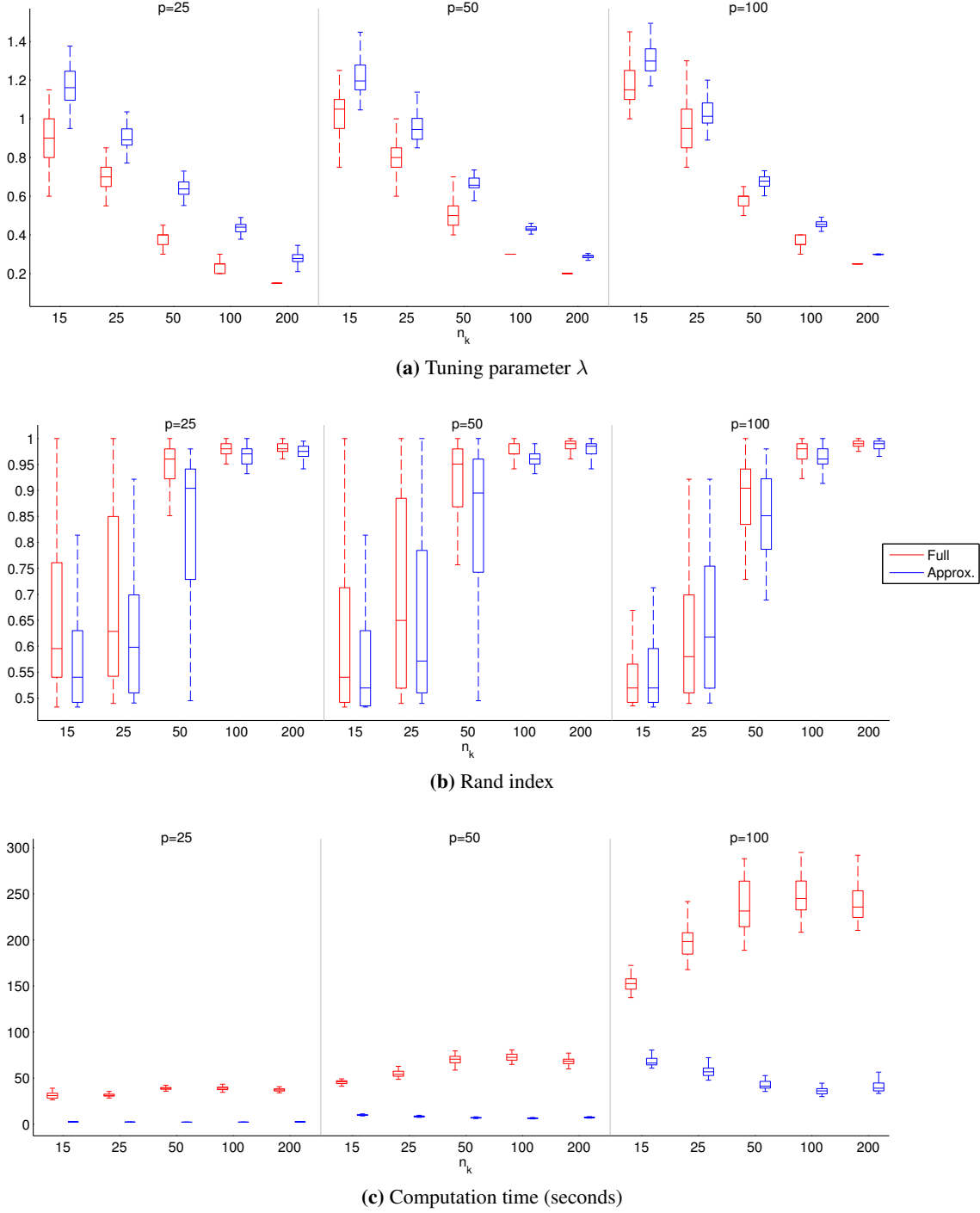
We presented a study of model-based clustering with mixtures of  $\ell_1$ -penalized Gaussian graphical models. Penalization has emerged as a key approach in high-dimensional statistics. However, choice of penalization set-up and the setting of tuning parameters can be non-trivial. We found that performance is dependent on choice of penalty term and method for setting the tuning parameter. Along with the standard  $\ell_1$  penalty ( $\gamma = 0$  in (6)) we considered an alternative penalty term, following recent work in penalized finite mixture of regression models (Khalili and Chen, 2007; Städler *et al.*, 2010), that is dependent on the mixing proportions  $\pi_k$  ( $\gamma = 1$  in (6)).

From our simulation study and application to breast cancer data, we draw some broad conclusions and recommendations, as follows. The combination of the  $\gamma = 1$  penalty term (incorporating mixing proportions), together with the BIC criterion for selecting the tuning parameter (regime B1), appears to provide the most accurate clustering and estimation of graphical model structure. The only exception is in settings where both dimensionality and sample size are small; here, the smaller tuning parameter values selected by train/test (or cross-validation) provide superior results (regimes T1/CV1). For estimation of the precision matrix itself (as opposed to estimation of sparsity structure only), we again recommend the penalty term with  $\gamma = 1$  and find that the less sparse estimates provided by train/test (or cross-validation) provide slight gains over BIC, except where dimensionality is large relative to sample size.

**Table 3:** Simulated data; precision matrix estimation. Elementwise  $\ell_1$  matrix norm shown for the methods and regimes in Table 1 at varying data dimensions  $p$  and per-cluster sample sizes  $n_k$  (smaller values indicate better agreement between true and inferred precision matrices). K-means clustering was followed by  $\ell_1$ -penalized estimation of Gaussian graphical model structure with penalty parameter set by train/test ('KM+T') or BIC ('KM+B'). For each  $(p, n_k)$  combination, the regime with lowest mean norm is highlighted in bold. (Mean matrix norm over 50 simulated datasets per  $(p, n_k)$  regime, standard deviations in parentheses; non-penalized, full covariance Gaussian mixture model (NP) could not be used for small sample sizes  $n_k \leq p$  due to invalid covariance estimates.)

p	$n_k$	T0	B0	T1	B1	Th	Bh	Ah	KM+T	KM+B	NP
25	15	64.00 (4.00)	65.73 (2.52)	<b>56.88</b> (4.87)	58.92 (3.12)	60.77 (5.59)	59.71 (3.07)	69.21 (2.58)	62.05 (4.35)	158.01 (136.36)	-
	25	64.41 (4.38)	66.39 (2.53)	<b>52.36</b> (4.47)	56.64 (3.61)	57.79 (6.48)	57.35 (4.01)	67.31 (3.63)	60.02 (4.35)	62.92 (31.35)	-
	50	47.56 (9.87)	57.83 (11.67)	47.76 (6.83)	49.89 (4.01)	<b>44.06</b> (2.45)	47.51 (3.36)	64.50 (3.28)	60.15 (6.37)	56.20 (4.22)	814.79 (282.61)
	100	35.53 (1.86)	37.73 (7.06)	35.43 (1.88)	38.22 (2.42)	<b>35.15</b> (1.95)	38.04 (2.35)	61.41 (3.50)	62.78 (8.74)	54.86 (5.87)	275.75 (29.26)
	200	<b>26.56</b> (1.48)	27.81 (3.22)	27.86 (2.56)	29.73 (1.59)	27.57 (2.52)	29.64 (1.66)	56.68 (3.18)	64.78 (11.49)	53.09 (7.38)	112.64 (21.30)
	15	126.47 (5.57)	129.52 (5.09)	119.20 (6.54)	<b>117.25</b> (5.25)	123.93 (9.17)	119.21 (5.60)	137.91 (5.81)	124.41 (6.25)	450.23 (377.88)	-
	25	129.08 (4.89)	129.75 (4.46)	118.52 (11.53)	<b>113.23</b> (5.46)	123.89 (11.13)	114.08 (5.54)	136.01 (5.25)	124.50 (6.53)	159.21 (172.28)	-
	50	128.38 (7.68)	129.29 (3.57)	<b>94.08</b> (4.02)	103.06 (6.16)	102.96 (10.57)	102.88 (5.00)	134.11 (5.76)	123.51 (6.45)	112.27 (4.56)	-
	100	120.24 (14.90)	125.51 (7.66)	78.95 (3.26)	82.77 (3.12)	<b>77.24</b> (3.56)	82.90 (3.51)	127.33 (7.37)	121.14 (7.77)	104.06 (4.46)	1366.07 (96.11)
	200	63.69 (2.70)	63.79 (2.95)	<b>63.67</b> (2.69)	67.31 (2.44)	64.10 (2.73)	67.15 (2.46)	119.91 (8.74)	122.92 (10.50)	98.90 (5.39)	587.84 (23.48)
100	15	250.78 (8.60)	256.29 (8.09)	247.53 (13.87)	<b>236.25</b> (10.02)	249.46 (14.67)	237.79 (9.50)	277.45 (9.77)	262.41 (10.77)	799.86 (794.03)	-
	25	256.42 (7.57)	256.42 (7.66)	249.49 (14.25)	<b>229.69</b> (9.76)	253.52 (12.55)	231.93 (8.59)	277.61 (9.50)	259.05 (11.26)	427.52 (534.15)	-
	50	254.02 (8.77)	249.76 (6.96)	<b>203.44</b> (16.47)	205.45 (6.45)	248.40 (23.35)	208.04 (7.93)	267.44 (10.53)	244.39 (10.51)	216.46 (9.12)	-
	100	269.91 (9.40)	246.16 (8.07)	172.36 (6.88)	176.36 (6.08)	<b>167.65</b> (5.26)	176.92 (6.43)	263.57 (12.30)	238.25 (12.31)	203.45 (9.27)	-
	200	171.59 (11.72)	239.45 (6.33)	158.82 (17.70)	144.53 (4.69)	<b>134.00</b> (3.76)	144.68 (4.31)	252.23 (15.93)	235.31 (19.05)	183.98 (8.16)	3468.41 (170.53)





**Figure 3: Simulated data; heuristic approach for tuning parameter selection.** (a) Boxplots over the tuning parameters selected by the heuristic method (see text for details) under regime B1 (mixture model with  $\gamma = 1$  and BIC) are shown (blue boxes), together with the corresponding values obtained with the full, non-approximate approach (red boxes). (b) Resulting Rand indices and (c) computational time required to set the parameter are also shown. (All results are over 50 simulated datasets for each  $(p, n_k)$  regime).

The deleterious effect of the standard  $\ell_1$  penalty term ( $\gamma = 0$ ), at all but the largest sample sizes, is intriguing. As described above, this is due to the fact that the standard penalty term leads to cluster-specific penalties in the EM update for the precision matrices. (Indeed, we observed similar results when setting cluster-specific penalties analytically in a non-mixture model setting). These cluster-specific penalties are inversely proportional to the mixing proportions  $\pi_k$ : in itself this behavior seems intuitively appealing since clusters with small effective sample sizes are then more heavily regularized. However, we observed that a substantially higher penalty is applied to one cluster over the other, indicating that samples were mostly being assigned to the same cluster. This is likely due to the ‘unpopular’ cluster having a poor precision matrix estimate due to a large penalty. We note that this behavior is not due to (lack of) EM convergence; the penalized likelihood scores from these incorrect clusterings were higher than those obtained using the true cluster labels.

The related non-mixture model approach proposed by Mukherjee and Hill (2011) also performed well in our studies, but clustering results (both from simulated and real data) indicate that a mixture model with EM (and  $\gamma = 1$  in the penalty term) offers more robust results.

Although the approaches we recommend performed well in the examples we considered, sensitivity to penalty formulation and the setting of tuning parameters remain a concern for penalized mixtures. Further work will be needed to better understand how such approaches behave in other settings and in higher dimensions. Cluster-specific scaling could also pose difficulties for penalization, as discussed recently in the context of hidden Markov models in Städler and Mukherjee (2012), who propose penalisation using the inverse correlation matrix as a potential solution. The approach proposed here could be adapted to use inverse correlation in place of inverse covariance.

We note that while for simplicity and tractability we focused on the  $K = 2$  clusters case, the methods we discuss are immediately applicable to the general  $K$ -cluster case. Moreover, since the approach we propose is model-based, established approaches for model selection in clustering, including information criteria, train/test and cross-validation, can be readily employed to select or explore  $K$ .

Our results demonstrate the necessity of some form of regularization to enable the use of Gaussian graphical models for clustering in settings of moderate-to-high dimensionality; indeed, we see clear benefits of penalization already in the  $p = 25$  case. The  $\ell_1$ -penalty is an attractive choice since it encourages sparsity at the level of graphical model structure, and estimation with the graphical lasso algorithm (Friedman *et al.*, 2008) is particularly efficient, which is important in the clustering setting, where multiple iterations are required. Alternatives include shrinkage estimators (Schäfer and Strimmer, 2005) and Bayesian approaches (Dobra *et al.*, 2004; Jones *et al.*, 2005). However, it has been shown that the  $\ell_1$ -penalized precision matrix estimator (2) is biased (Lam and Fan, 2009). Alternative penalties have been proposed in a regression setting to ameliorate this issue; the non-concave SCAD penalty (Fan and Li, 2001) and adaptive  $\ell_1$  penalty (Zou, 2006), and have recently been applied to sparse precision matrix estimation (Fan *et al.*, 2009). These penalties are generally computationally more intensive, but it remains an open question whether they improve clustering accuracy relative to the  $\ell_1$  penalty considered here.

Graphical models based on direct acyclic graphs (DAGs) are frequently used for network inference, especially in biological settings where directionality may be meaningful (for example, Friedman *et al.*, 2000; Husmeier, 2003; Perrin *et al.*, 2003; Mukherjee and Speed, 2008; Ellis and Wong, 2008; Hill *et al.*, 2012). A natural extension to the ideas discussed here would be to develop a clustering approach based on DAGs rather than undirected models.

There are several recent and attractive extensions to graphical Gaussian model estimation that

could be exploited to improve and extend the methods we discuss. For example, the time-varying Gaussian graphical model approach of Zhou *et al.* (2010) could be employed, or prior knowledge of network structure could be taken into account (Anjum *et al.*, 2009); such information is abundantly available in biological settings. The joint estimation method for Gaussian graphical models proposed by Guo *et al.* (2011) explicitly models partial agreement between network structures corresponding to *a priori* known clusters. Such partial agreement could be incorporated in the current setting where clusters are not known *a priori*.

**Acknowledgement:** We thank P. T. Spellman, N. Städler and N. Meinshausen for discussions. Supported by EPSRC EP/E501311/1 and NCI U54 CA 112970 and the Cancer Systems Biology Center grant from the Netherlands Organisation for Scientific Research.

## References

- Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., Powell, J. I., Yang, L., Marti, G. E., Moore, T., Hudson, J., Lu, L., Lewis, D. B., Tibshirani, R., Sherlock, G., Chan, W. C., Greiner, T. C., Weisenburger, D. D., Armitage, J. O., Warnke, R., Levy, R., Wilson, W., Grever, M. R., Byrd, J. C., Botstein, D., Brown, P. O., and Staudt, L. M. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, **403**, 503–511.
- Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA*, **96**, 6745–6750.
- Altay, G., Asim, M., Markowetz, F., and Neal, D. (2011). Differential C3NET reveals disease networks of direct physical interactions. *BMC Bioinf.*, **12**, 296.
- Anjum, S., Doucet, A., and Holmes, C. C. (2009). A boosting approach to structure learning of graphs with and without prior knowledge. *Bioinformatics*, **25**, 2929–2936.
- Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F., and Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, **16**, 412–424.
- Banerjee, O., El Ghaoui, L., and D’Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *J. Mach. Learn. Res.*, **9**, 485–516.
- Cai, T., Liu, W., and Luo, X. (2011). A constrained  $\ell_1$  minimization approach to sparse precision matrix estimation. *J. Am. Stat. Assoc.*, **106**, 594–607.
- Chuang, H.-Y., Lee, E., Liu, Y.-T., Lee, D., and Ideker, T. (2007). Network-based classification of breast cancer metastasis. *Mol. Syst. Biol.*, **3**, 140.
- D’Aspremont, A., Banerjee, O., and El Ghaoui, L. (2008). First-order methods for sparse covariance selection. *SIAM J. Matrix Anal. Appl.*, **30**, 56–66.
- Datta, S. and Datta, S. (2003). Comparisons and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics*, **19**, 459–466.

- de Souto, M., Costa, I., de Araujo, D., Ludermir, T., and Schliep, A. (2008). Clustering cancer gene expression data: a comparative study. *BMC Bioinf.*, **9**, 497.
- Dempster, A. P. (1972). Covariance selection. *Biometrics*, **28**, 157–175.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. B*, **39**, 1–38.
- Dobra, A., Hans, C., Jones, B., Nevins, J., Yao, G., and West, M. (2004). Sparse graphical models for exploring gene expression data. *J. Multivar. Anal.*, **90**, 196–212.
- Edwards, D. (2000). *Introduction to Graphical Modelling*. Springer, New York.
- Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, **95**, 14863–14868.
- Ellis, B. and Wong, W. H. (2008). Learning causal Bayesian network structures from experimental data. *J. Am. Stat. Assoc.*, **103**, 778–789.
- Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Stat. Assoc.*, **96**, 1348–1360.
- Fan, J., Feng, Y., and Wu, Y. (2009). Network exploration via the adaptive lasso and scad penalties. *Ann. Appl. Stat.*, **3**, 521–541.
- Fraley, C. and Raftery, A. E. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *Comput. J.*, **41**, 578–588.
- Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, **9**, 432–441.
- Friedman, N., Linial, M., Nachman, I., and Pe’er, D. (2000). Using Bayesian networks to analyze expression data. *J. Comp. Bio.*, **7**, 601–620.
- Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.
- Guo, J., Levina, E., Michailidis, G., and Zhu, J. (2011). Joint estimation of multiple graphical models. *Biometrika*, **98**, 1–15.
- Hecker, M., Lambeck, S., Toepfer, S., van Someren, E., and Guthke, R. (2009). Gene regulatory network inference: Data integration in dynamic models - a review. *Biosystems*, **96**, 86–103.
- Hill, S. M., Lu, Y., Molina, J., Heiser, L. M., Spellman, P. T., Speed, T. P., Gray, J. W., Mills, G. B., and Mukherjee, S. (2012). Bayesian inference of signaling network topology in a cancer cell line. *Bioinformatics*, **28**, 2804–2810.
- Husmeier, D. (2003). Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, **19**, 2271–2282.

- Jones, B., Carvalho, C., Dobra, A., Hans, C., Carter, C., and West, M. (2005). Experiments in stochastic computation for high-dimensional graphical models. *Stat. Sci.*, **20**, 388–400.
- Kerr, G., Ruskin, H. J., Crane, M., and Doolan, P. (2008). Techniques for clustering gene expression data. *Comput. Biol. Med.*, **38**, 283–293.
- Khalili, A. and Chen, J. (2007). Variable selection in finite mixture of regression models. *J. Am. Stat. Assoc.*, **102**, 1025–1038.
- Lam, C. and Fan, J. (2009). Sparsistency and rates of convergence in large covariance matrix estimation. *Ann. Stat.*, **37**, 4254–4278.
- Lee, W.-P. and Tzou, W.-S. (2009). Computational methods for discovering gene networks from expression data. *Briefings Bioinf.*, **10**, 408–423.
- Madeira, S. C. and Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **1**, 24–45.
- Markowetz, F. and Spang, R. (2007). Inferring cellular networks - a review. *BMC Bioinf.*, **8**, S5.
- McLachlan, G. J., Bean, R. W., and Peel, D. (2002). A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, **18**, 413–422.
- Meinshausen, N. and Bühlmann, P. (2006). High dimensional graphs and variable selection with the Lasso. *Ann. Stat.*, **34**, 1436–1462.
- Mukherjee, S. and Hill, S. M. (2011). Network clustering: Probing biological heterogeneity by sparse graphical models. *Bioinformatics*, **27**, 994–1000.
- Mukherjee, S. and Speed, T. P. (2008). Network inference using informative priors. *Proc. Natl. Acad. Sci. USA*, **105**, 14313–14318.
- Needham, C. J., Bradford, J. R., Bulpitt, A. J., and Westhead, D. R. (2007). A primer on learning in Bayesian networks for computational biology. *PLoS Comput. Biol.*, **3**, e129.
- Pan, W. and Shen, X. (2007). Penalized model-based clustering with application to variable selection. *J. Mach. Learn. Res.*, **8**, 1145–1164.
- Pe’er, D. and Hachohen, N. (2011). Principles and strategies for developing network models in cancer. *Cell*, **144**, 864–873.
- Perrin, B.-E., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J., and d’Alche Buc, F. (2003). Gene networks inference using dynamic Bayesian networks. *Bioinformatics*, **19**, ii138–ii148.
- Rothman, A. J., Bickel, P. J., Levina, E., and Zhu, J. (2008). Sparse permutation invariant covariance estimation. *Electron. J. Statist.*, **2**, 494–515.
- Schäfer, J. and Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Stat. Appl. Genet. Mol. Biol.*, **4**, 32.
- Segal, E., Shapira, M., Regev, A., Pe’er, D., Botstein, D., Koller, D., and Friedman, N. (2003). Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat. Genet.*, **34**, 166–176.

- Städler, N. and Mukherjee, S. (2012). Penalized estimation in high-dimensional hidden Markov models with state-specific graphical models. Pre-print. arXiv:1208.4989.
- Städler, N., Bühlmann, P., and van de Geer, S. (2010).  $\ell_1$ -penalization for mixture regression models. *TEST*, **19**, 209–256.
- Thalamuthu, A., Mukhopadhyay, I., Zheng, X., and Tseng, G. C. (2006). Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics*, **22**, 2405–2412.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. B*, **58**, 267–288.
- Toh, H. and Horimoto, K. (2002). Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling. *Bioinformatics*, **18**, 287–297.
- Yuan, M. and Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*, **94**, 19–35.
- Zhou, H., Pan, W., and Shen, X. (2009). Penalized model-based clustering with unconstrained covariance matrices. *Electron. J. Statist.*, **3**, 1473–1496.
- Zhou, S., Lafferty, J., and Wasserman, L. (2010). Time varying undirected graphs. *Mach. Learn.*, **80**, 295–319.
- Zou, H. (2006). The adaptive lasso and its oracle properties. *J. Am. Stat. Assoc.*, **101**, 1418–1429.